# Math 261C: Randomized Algorithms

Lecture topic: An Application of PIT and Interactive Proofs

Lecturer: Sam Buss
Scribe notes by: Christian Woods
Date: May 16, 2014

## 1. An Application of Polynomial Identity Testing

**Definition.** *Let $G$ be a bipartite graph with parts $U$ and $V$ such that $|U| = |V| = n$. A perfect matching of $G$ is a subset of the edges of $G$ such that the induced graph on these edges is 1-regular.*

Problem: Given such a bipartite $G$, does it have a perfect matching?

Edmonds gives a deterministic polynomial time algorithm for solving this problem based on Hall's Marriage Theorem [1]. We will analyze a randomized algorithm based on Polynomial Identity Testing (PIT).

Given $G$, form the matrix $A = (a_{ij})_{i,j=1}^{n}$, where

$$a_{ij} = \begin{cases} x_{ij} & \text{if there is an edge from the } i\text{th vertex in } U \text{ to the } j\text{th vertex in } V \\ 0 & \text{otherwise,} \end{cases}$$

where the $x_{ij}$ are distinct variables.

**Theorem [Edmonds].** *$G$ has a perfect matching if and only if $\det(A) \neq 0$.*

*Proof.* By definition we have

$$\det(A) = \sum_{\pi \in S_n} \operatorname{sgn}(\pi) a_{1\pi(1)} a_{2\pi(2)} \cdots a_{n\pi(n)},$$

where $\pi$ ranges over the permutations of $\{1, 2, \ldots, n\}$.

But a term in this sum is nonzero if and only if $\pi$ defines a perfect matching in $G$. Moreover, nonzero terms will not cancel out. The theorem follows. □

But the determinant of $A$ is computable by a polynomial size straight-line program (if we use the cofactor method to calculate it). Therefore we have the following algorithm:

**Input**: $G$ a bipartite graph with equipotent parts.
**Output**: If $G$ has a perfect matching.
Form the matrix $A$ from $G$.
Compute $\det(A)$ using a straight-line program.
Use the randomized PIT algorithm to check if $\det(A) \equiv 0$.
**if** $\det(A) \equiv 0$ **then**
| return "No.
**else**
| return "Yes."
**end**

**Algorithm 1:** Determining if $G$ has a perfect matching.

## 2. Interactive Proofs

We have been talking about randomized algorithms giving strong "proof" of facts like "$f(x) \equiv 0$" or "$p$ is prime."

But these are not proofs we can just write down - they require we perform random sampling. We might say these are not "static" proofs, because active computation is a necessary step for the "reader".

We will generalize these ideas to "interactive randomized proofs."

Example:

The Graph Isomorphism Problem (GI)

Input: A pair of graphs $G_1$ and $G_2$ on the same number of vertices.

Goal: Be convinced that $G_1 \cong G_2$, if this statement is true.

The traditional static proof for GI is to provide an isomorphism $\phi : G_1 \to G_2$.

The Graph Nonisomorphism Problem (GNI)

Input: A pair of graphs $G_1$ and $G_2$ on the same number of vertices. Goal: Be convinced that $G_1 \ncong G_2$, if this statement is true.

There is no known general way to give static proofs of graph non-isomormorphism. In other words, it is an open problem whether or not GNI $\in NP$. Instead we investigate an interactive proof.

Interactive Protocol: In an interactive proof protocol there are two participants: (i) the verifier V, who wants to be convinced that $G_1 \ncong G_2$, and (ii) the prover P, who is all-knowing and all-powerful and wants to convince us that $G_1 \ncong G_2$. However, the proved may be malicious in that he may try to fool us. V is restricted to performing randomized polynomial time computations, whereas P is not.

**Data**: Graphs $G_1, G_2$ on vertices $\{1, 2, \ldots, n\}$ given to V and P
Step (1): V picks at random (i) $i \in \{1, 2\}$ (ii) a permutation $\pi \in S_n$.
V sends $K = \pi(G_i)$ to P, but keeps $i$ and $\pi$ secret.
Step (2): P picks a value $i' \in \{1, 2\}$, e.g. one such that $G_{i'} \cong K$.
P returns $i'$ to V.
Step (3): V accepts if $i = i'$.
V rejects if $i \neq i'$.

**Claims.** *(i) If $G_1 \ncong G_2$ then there exists a prover P such that Pr[V accepts] $= 1$.*

*(ii) If $G_1 \cong G_2$ then for all provers P, Pr[V accepts] $\leq \frac{1}{2}$.*

*Proof.* (i) is trivial, as P honestly determines which of $G_1$ and $G_2$ is isomorphic to $K$.

Now let $i' = p(K, G_1, G_2)$, where $p$ is the function computed by P. Note that

$$Pr[i = 1|K] = \frac{1}{2} = Pr[i = 2|K].$$

So

$$Pr[p(K, G_1, G_2) = i|K] = \frac{1}{2}.$$

□

By repeating this protocol $m$ times, we can get the probabilities to be 1 and $\frac{1}{2^m}$ respectively, instead of 1 and $\frac{1}{2}$.

This was an example of an IP (interactive proof) protocol. It uses "private coins," i.e. V is allowed to choose random values and hide them from P.

We could consider instead a "public coin" model. Now the verifier does not get to keep any secrets. These are called "Arthur-Merlin" protocols. We now call the verifier "Arthur," or A, and the prover "Merlin," or M. Arthur has *no* secrets from Merlin. We now turn this into a proper mathematical model.

**Definition.** *For $k \geq 2$ an even integer, an AM[k] protocol is a protocol such that*

*(i) the input $x$ has length $|x| = n$*

*(ii) For some polynomial $q(n)$, Arthur chooses the value $r_i \in \{0, 1\}^{q(n)}$ at random at the end of round $i$, for $i = 0, \ldots, \frac{k}{2}$*

*(iii)For some function $m$, Merlin computes $m_i = m(x, \langle r_0, \ldots, r_{i-1}\rangle)$ during round $i$, where $|m_i|$ is polynomial in $n$ for each $i = 1, \ldots, \frac{k}{2}$.*

*(iv) A has a polynomial time function out such that*

$$out(x, \langle r_0, \ldots, r_{\frac{k}{2}}\rangle, \langle m_1, \ldots, m_{\frac{k}{2}}\rangle) = \begin{cases} 1 & \text{if } A \text{ accepts} \\ 0 & \text{if } A \text{ rejects.} \end{cases}$$

*We say that a language $L$ belongs to $AM[k]$ if and only if for all $x \in \{0,1\}^*$ we have*

$$x \in L \implies Pr[A \text{ accepts}] \geq \frac{2}{3} \text{ (“completeness”)}$$

*and*

$$x \notin L \implies Pr[A \text{ accepts}] \leq \frac{1}{3} \text{ (“soundness”).}$$

For example, an $AM[6]$ protocol looks like

- A picks $r_0$
- M returns $m_1 = m(x, \langle r_0 \rangle)$
- A picks $r_1$
- M returns $m_2 = m(x, \langle r_0, r_1 \rangle)$
- A picks $r_2$
- M returns $m_3 = m(x, \langle r_0, r_1, r_2 \rangle)$.
- A picks $r_3$ and computes $out(x, \langle r_0, r_1, r_2, r_3 \rangle, \langle m_1, m_2, m_3 \rangle)$

By running multiple copies of an $AM[k]$ protocol in parallel and using the majority result as the answer, we can replace $\frac{2}{3}$ and $\frac{1}{3}$ with $1 - \frac{1}{2^m}$ and $\frac{1}{2^m}$ using the Chernoff bounds. (We do this by running $O(m)$ many copies of the protocol at once.)

For instance, $L \in AM[2]$ means

- A picks $r_0$
- M returns $m_1 = m(x, \langle r_0 \rangle)$
- A picks $r_1$ and computes $out(x, \langle r_0, r_1 \rangle, \langle m_1 \rangle)$

Next time we will show

**Theorem.** $GNI \in AM[2]$.

More generally, for $k \geq 2$ an even integer, $AM[k] \subseteq AM[2]$.

If $IP[k]$ is defined similarly for interactive proof protocols, then $IP[k] \subseteq AM[2]$.

Moreover we can achieve “perfect completeness” where

$$Pr[\text{A accepts}] = 1 \text{ for all } x \in L.$$

**Definition.** $IP = \bigcup_{c \in \mathbb{N}} IP[n^c]$.

**Definition.** $AM = AM[2]$.

**Theorem.** $IP = \bigcup_{c \in \mathbb{N}} AM[n^c]$.

## References

[1] Edmonds, Jack. "Maximum matching and a polyhedron with 0,1-vertices". Journal of Research National Bureau of Standards Section. 1965. B 69: 125-130.